

Throughput Optimal Random Medium Access Control for Relay Networks with Time-Varying Channels

Mehdi Salehi Heydar Abad, Ozgur Ercetin, Eylem Ekici

Abstract—The use of existing network devices as relays has a potential to improve the overall network performance. In this work, we consider a two-hop wireless relay setting, where the channels between the source and relay nodes to the destination node are time varying. The relay nodes are able to overhear the transmissions of the source node which may have a weak connection to the destination, and they help the source node by forwarding its messages to the destination on its behalf, whenever this is needed. We develop a distributed scheme for relay selection and channel access that is suitable for time-varying channels, and prove that this scheme is throughput optimal. We obtain the achievable rate region of our proposed scheme analytically for a relay network with a single source and a single relay node. Meanwhile, for a more general network with more than one relay nodes, we perform Monte-Carlo simulations to obtain the achievable rate region. In both cases, we demonstrate that the achievable rate region attained with our distributed scheme is the same as the one attained with centralized optimal scheme.

Index Terms—Relay networks, throughput optimal, distributed.

I. INTRODUCTION

A. Motivation

In wireless networks, there is always a node which has the worst channel quality (e.g., cell edge node) due to its location, shadowing effects or physical radio capabilities. Wireless channel quality for these nodes may become so weak that in order to maintain a reliable communication, some relays have to be employed on the path between the source and the destination node to aid their communication [1]. Instead of deploying separate devices solely used for relaying, it is possible to utilize the existing wireless devices in the network for the same purpose. It is expected that future wireless devices are going to be highly adaptive and flexible, and thus, opening up new paradigms where they can adapt themselves utilizing any network resource, e.g., as other peer nodes in their vicinity, to improve their communications.

One of such paradigms is relay networks where the aim is to improve the network throughput by allowing wireless nodes to participate in the transmission, when they are neither the intended destination nor the source [2]. A widely used wireless relay network model consists of two hops, i.e., there

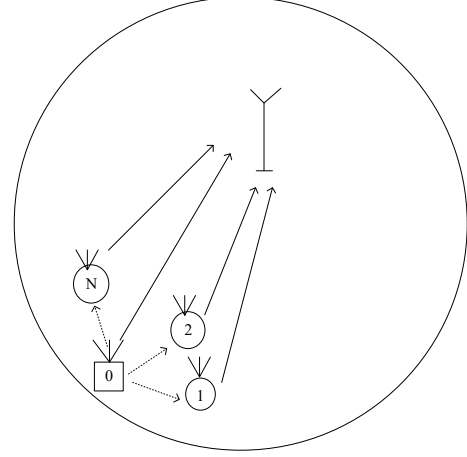


Fig. 1: Relay Network.

are a source, a destination, and multiple relay nodes [3] as in Figure 1. The basic idea is that the relay nodes, overhearing the message transmitted from the source node, forward the same message to the destination node instead of treating it as interference [4].

Fundamentally, this idea of relaying is very similar to routing in multi-hop networks, and there is already a rich literature on the subject. In particular, centralized *throughput optimal* policies were developed for opportunistic multi-hop relay networks in [5]–[7]. A policy is *throughput optimal* in the sense of having an achievable rate region that coincides with the network achievable rate region, and thus, a superset of the achievable rate region of any other policy. However, a centralized policy requires global information of the entire network. The collection of this information is costly in terms of power consumption and time. On the other hand, distributed (decentralized) policies may alleviate these implementation costs by running a loosely coordinated scheme at each network node.

A common network setting is where the source and relay nodes are located in such a way that the source is *furthest* to the destination, while the relays are relatively *closer*. This notion of distance includes both real distances in free-space communication, or virtual distances that also include the effects of fading and shadowing [8]. Hence, the source has the worst channel quality in terms of probability of successfully transmitting its packets to the destination.

There is a number of works developing distributed policies

M.S.H. Abad and O. Ercetin are with the Faculty of Engineering and Natural Sciences, Sabanci University, 34956 Istanbul, Turkey.

E. Ekici is with the Department of Electronics and Computer Engineering, The Ohio State University, Columbus, OH.

This work is supported in part by a grant from European Commission MC-IRSES programme.

for two-hop wireless relay networks with the objective of improving the transmission rate of the source node [9]–[12]. In these works, the time-varying wireless channel is modeled as an ON-OFF random process for analytical tractability, where a packet is successfully decoded at the destination node when the channel is ON or it is lost when the channel is OFF. Note that modeling the channel as an ON-OFF random process is an approximation of a more general time-varying channel model, where the channel is a multi-state process, with each state supporting a maximum transmission rate with which the data can be reliably sent [13]. The authors then propose *suboptimal* channel access algorithms, and calculate the achievable transmission rate of the source node by employing Loynes’ theorem [14].

Unlike prior works, we develop a throughput optimal distributed policy called Relay Q-CSMA (RQ-CSMA), for relay networks with time-varying channels, and show that its achievable rate region coincides with the achievable rate region of a centralized optimal policy. Our policy achieves this without the knowledge of the channel statistics, and it performs the scheduling of the source and relay nodes by only utilizing local information without any explicit message exchanges.

B. Related Work

The network setting used in this paper was previously studied under different contexts and assumptions in the literature. In particular, for multi-hop opportunistic routing it is shown that Maximum Weight Scheduling (MWS) algorithm, which is a centralized scheme is throughput optimal [5]. MWS and many of its variants exist in the literature addressing the optimal resource allocation under different network assumptions and for different applications, e.g., cognitive networks [15], relay networks [16], etc [13], [17], [18]. Unfortunately, MWS is known to be NP-hard for general networks [19], [20] and it is not amenable to a distributed implementation.

The high time complexity of the centralized algorithms such as MWS was addressed in the literature by developing low-complexity sub-optimal algorithms. In particular, maximal scheduling is a low-complexity alternative to MWS that is amenable to parallel and distributed implementation [21]. However, maximal scheduling may only achieve a fraction of the achievable rate region [22]. Greedy Maximal Scheduling (GMS), also known as Longest-Queue-First (LQF), is another low-complexity alternative to MWS [23] with a complexity that grows linearly with the total number of the links [24]. Its performance has been observed to be close to optimum in a variety of wireless network scenarios [25]. Although these algorithms reduce the complexity of MWS, they have two main shortcomings. First, they require network wide queue length information exchange, and second, they are mainly suitable for links with time-invariant properties.

Another class of distributed scheduling policies, called Queue-length-based Random Access Scheduling policies, use local message exchanges to resolve the contention problem [26], [27]. By adjusting each link’s contention probability using the link’s local queue information, it provides explicit tradeoffs between efficiency, complexity, and the

contention period. Carrier-Sensing-Multiple-Access (CSMA)-based scheduling policies [28], [29], reduce the complexity by simplifying the comparison process, by exploiting carrier-sensing. Nonetheless, these results indicate that good throughput performance may be attained for time invariant channels using algorithms with very low complexity.

In practice, however, most wireless systems have time varying characteristics. When link states vary over time, the system throughput can be improved by scheduling those links with better states. This is known as the opportunistic gain [30]. However, many of the low-complexity scheduling algorithms in the literature do not exploit the opportunistic gain, and their performance in time varying channels are shown to be significantly lower [31]–[33].

Recently, a few other low-complexity schemes have been proposed that are provably efficient over time variable channels [31], [32]. However, these algorithms, depending on the network structure, require either local or global queue length and channel state information (CSI) exchange. In Opportunistic ALOHA [34], [35] the transmission probability is allowed to be a function of the channel state information and maximum throughput of the system is achieved. Distributed Opportunistic Scheduling (DOS) [36], [37] involves a process of joint channel probing and distributed scheduling. The authors show that the optimal scheme for DOS turns out to be a pure threshold policy, where the rate threshold can be obtained by solving a fixed-point equation. However both Opportunistic ALOHA and DOS are designed under the assumption of saturated queues and their performance have no guarantees for unsaturated queues.

The use of relay nodes has also been investigated in the context of cellular networks. In [38] a throughput optimal centralized downlink scheduling policy is developed for multi-hop relaying in a cellular network. Note that the uplink scheduling is more challenging, since the queue length information is not readily available at the base station. Conventional cellular architecture treats the mobile users as simple transceivers and due to this rigid client-server scheme, they are fully commanded by cellular base stations. In future cellular architectures, mobile users will be given more freedom, and thus, they will be treated as local micro-operators to improve network coverage [39]. Device-to-Device (D2D) communication is another paradigm which appears to be a promising component in the next generation cellular networks [40]. D2D communication was proposed in [41] to enable multi-hop relaying in cellular networks. The potential of D2D communications for improving spectral efficiency of cellular networks was also identified in [42], [43]. There are numerous distributed D2D policies [44]–[47], addressing the data relay services enabled by existing devices in the network. However, most prior works in the literature ignore the network stability and do not investigate throughput optimality.

C. Main results and organization

In this work, we take both channel state and queue length information into account when scheduling the source and relay nodes. We propose a distributed CSMA based scheduling

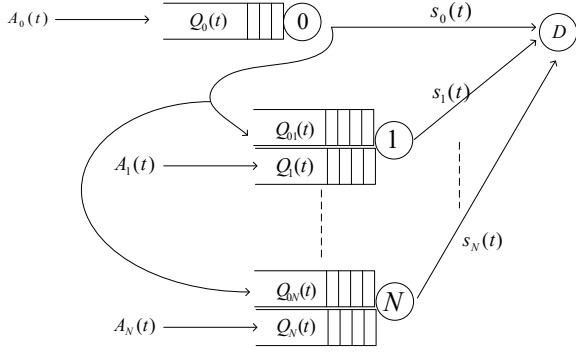


Fig. 2: Relay Network. Channels between nodes and destination are ON-OFF whereas the channels between all pairs of nodes are always ON.

algorithm which is analytically shown to be the throughput optimal. The time complexity of the algorithm is shown to be linear in the number of nodes. The rest of the paper is organized as follows. In Section II, we introduce the relay network model. In Section III, we give a brief introduction to MWS and later propose our scheduling and contention resolution algorithms. In Section IV, we analytically prove throughput optimality of our algorithm. We establish the achievable rate region for a source and a single relay node in Section V and finally, we evaluate the performance of different scheduling algorithms via simulations in Section VI.

II. SYSTEM MODEL

In this section, we introduce the details of our system model and the definitions of the main concepts that are used in relation to this model. Throughout the paper, the following notations are adopted. Upper case and lower case bold symbols denote matrices and vectors, respectively. Upper case calligraphic symbols denote sets.

A. Network Model

We consider a wireless relay network consisting of a source node and N relay nodes all wishing to communicate with a common destination node as shown in Figure 2. We consider a time-slotted system where the time slot is the resource to be shared among different nodes. We adopt a non-interference model where only one node is transmitting at a given time¹. The length of the time slot is equal to the transmission time of a single packet.

We assume that the channels between $N + 1$ nodes and the destination node are time-varying ON-OFF channels, where a packet is either successfully decoded at the destination or lost with a certain probability. The nodes with better channel conditions to the destination have higher successful transmission probabilities. We will refer to the source node as node 0 and N relay nodes as node i , $i = 1, \dots, N$. We assume that relay nodes are in close proximity to node 0 so

that the channels between node 0 and nodes $i = 1, \dots, N$ are always ON and node 0 can always transmit successfully to any node $i = 1, \dots, N$. This assumption is realistic in practical scenarios (e.g., see [48]), due to proximity of source node and relays.

Let $s_i(t) \in \{0, 1\}$ be the channel state between node i , $i = 0, 1, \dots, N$ and the destination at time slot t . The random channel states, $s_i(t)$, are assumed to be independent and identically distributed (iid) across time and nodes. Let ρ_i be the probability that the channel between node i and the destination is ON, i.e., $s_i(t) = 1^2$.

Let $A_i(t)$ be the number of packets arriving to node $i = 0, 1, \dots, N$ at time t . The arrival processes are assumed to be iid across time and nodes. Let $\lambda_i = \mathbb{E}[A_i(t)]$ be the rate of arrival to node i . We also assume that $A_i(t) \leq A_{max}$ and $\mathbb{E}[A_i(t)^2]$ is finite for all i and t . The incoming packets are stored in a queue until they are finally transmitted.

Let $Q_i(t)$ be the size of the queue storing the packets of node $i = 0, 1, \dots, N$ at time t . Meanwhile, each node $i = 1, \dots, N$ also keeps a separate queue for node 0 packets that they are supposed to relay on its behalf. Let $Q_{0i}(t)$, $i = 1, 2, \dots, N$ be the size of this queue at time t .

Finally, we assume that the transmission rate of each node is the same and equal to one unit sized packet per time slot.

B. The Relaying Scheme

At the beginning of time slot t , the channel is acquired by one of the nodes according to the scheme to be explained in the subsequent section. If node 0 acquires the channel, it sends its packet to the destination if $s_0(t) = 1$, i.e., the channel between node 0 and destination is ON. Otherwise, it forwards its packet to one of the nodes $1, \dots, N$ for future delivery of the packet to the destination via the selected node i , $i = 1, 2, \dots, N$. Node 0 forwards the packet to node i^* , where $i^* = \arg \max_j \{Q_0(t) - Q_{0j}(t)\}$. Meanwhile, if a node other than node 0 acquires the channel in a time slot, it may transmit either one of its own packets or a packet received from node 0 in previous time slots. Node i , $i = 1, \dots, N$ transmits a packet from one of its two queues, i.e., $Q_{0i}(t)$ or $Q_i(t)$, with the largest backlog. The destination sends an Acknowledgment (ACK) message after the reception of the packet. ACK message is perfectly received by both node 0 and the transmitting node i , $i = 1, \dots, N$, and thus, node 0 is cognizant of the delivery of its packet by the node i , $i = 1, 2, \dots, N$. Note that the assumption of perfect reception of ACK/NACK is not necessary. It can be shown in a similar way as was done in [13] that as long as the queues are updated periodically with period $T > 1$, our results still hold.

Let $x(t) = \mathcal{I}, 0, 1, \dots, N$ be the scalar denoting the scheduled node for transmission, where \mathcal{I} denotes the *Idle* state indicating no nodes are scheduled for transmission. Based

¹This assumption is widely used in the literature, e.g., it is proposed that the cellular spectrum is utilized for both D2D and cellular communications (i.e., underlay inband D2D) [40].

² $\rho_i = \Pr \left[\frac{|h_i|^2 P}{N_0} > \gamma \right]$, where h_i is the channel gain from node i to the destination node, N_0 is the noise power level at the destination and γ is the Signal-to-Noise Ratio (SNR) threshold required for correct decoding.

on the aforementioned relaying scheme, the queues evolve as follows:

$$Q_0(t+1) = \max [Q_0(t) - \mathbb{1}_{\{x(t)=0\}}, 0] + A_0(t), \quad (1)$$

$$Q_i(t+1) = \max [Q_i(t) - \mathbb{1}_{\{x(t)=i\}} \cdot \mathbb{1}_{\{Q_i(t) \geq Q_{0i}(t)\}}, 0] + A_i(t), \quad \forall i = 1, \dots, N \quad (2)$$

$$Q_{0i^*}(t+1) = \max [Q_{0i^*}(t) - \mathbb{1}_{\{x(t)=i^*\}} \cdot \mathbb{1}_{\{Q_{i^*}(t) < Q_{0i^*}(t)\}}, 0] + (1 - s_0(t)) \mathbb{1}_{\{x(t)=0\}}, \quad \text{for } i^* = \arg \max_j \{Q_0(t) - Q_{0j}(t)\}, \quad (3)$$

$$Q_{0i}(t+1) = \max [Q_i(t) - \mathbb{1}_{\{x(t)=i\}} \cdot \mathbb{1}_{\{Q_i(t) < Q_{0i}(t)\}}, 0] \quad \forall i \neq i^*, \quad (4)$$

where $\mathbb{1}_{\{\cdot\}}$ is the indicator function, and its value is equal to 1 if the condition inside the bracket is true, and 0 otherwise.

III. DISTRIBUTED ALGORITHM

A. Background and Preliminaries

Before we proceed with the discussion of our proposed distributed scheduling algorithm, we give a brief overview of the well-known Maximum Weight Scheduling (MWS) [18] and Q-CSMA algorithms [29] in the context of our relay network.

Maximum Weight Scheduling: According to MWS algorithm, each node is assigned a weight that is the product of its *differential* queue length and the channel state. MWS algorithm chooses the node with the maximum weight for transmission at time slot t . The differential queue length of the node is the difference of the queue lengths of the ingress and egress nodes. Note that in our relay network, the destination is the sink node, and thus, its queue length is always zero. Let $Q_r(t)$ be the maximum differential backlog of node 0 with respect to nodes $1, \dots, N$ at time slot t , which is defined as

$$Q_r(t) = \max_{i=1,2,\dots,N} \{Q_0(t) - Q_{0i}(t), 0\}. \quad (5)$$

In our network model, we assign weights to nodes in the network, and aim to schedule the node (and the queue of that node) with the highest weight at every slot. The weight of node i at slot t , $\omega_i(t)$, is defined as follows:

$$\omega_0(t) = f_0(Q_0(t))s_0(t) + f_0(Q_r(t))(1 - s_0(t)), \quad (6)$$

$$\omega_i(t) = \max \{f_i(Q_{0i}(t)), f_i(Q_i(t))\} s_i(t), \quad \forall i \neq 0, \quad (7)$$

where $f_i : [0, \infty] \rightarrow [0, \infty]$, $i = 0, 1, \dots, N$ are functions that should satisfy the following conditions [13]:

- 1) $f_i(Q_i)$ is a non decreasing, continuous function with $\lim_{Q_i \rightarrow \infty} f_i(Q_i) = \infty$.
- 2) Given any $M_1 > 0$, $M_2 > 0$ and $0 < \epsilon < 1$, there exists a $Q < \infty$, such that for all $Q_i > Q$ and $\forall i$, we have

$$(1 - \epsilon)f_i(Q_i) \leq f_i(Q_i - M_1) \leq f_i(Q_i + M_2) \leq (1 + \epsilon)f_i(Q_i). \quad (8)$$

It can be shown that for this system model, the centralized MWS algorithm is throughput optimal using the Lyapunov drift theorem [18]. However, note that MWS requires the complete instantaneous queue length and channel state information.

Queue based Carrier-Sensing-Multiple-Access (Q-CSMA): Carrier-Sensing-Multiple-Access (CSMA)-based scheduling policies [28], [29], reduce the complexity of MWS by simplifying the comparison of the weights exploiting the carrier-sensing mechanism. In Q-CSMA, each time slot t is divided into a *contention* and a *data* slot. For any node i , let $\mathcal{C}(i)$ be the set of conflicting nodes (called conflict set) of node i , i.e., $\mathcal{C}(i)$ is the set of nodes such that if any one of them is active, then node i cannot be active. In the contention slot, the network first selects a set of nodes that do not conflict with each other, denoted by $\mathcal{D}(t)$. Note that this set of nodes is also a feasible schedule, but this is not the schedule to be used for data transmission. $\mathcal{D}(t)$ is called the *decision schedule* at time t .

The network selects a decision schedule according to a randomized procedure, i.e., it selects $\mathcal{D}(t) \in \mathcal{M}$ with probability $\alpha(\mathcal{D}(t))$, where $\sum_{\mathcal{D}(t) \in \mathcal{M}} \alpha(\mathcal{D}(t)) = 1$, and \mathcal{M} is the set of all possible feasible schedules of the network. Then, the scheduling procedure proceeds as follows: Each node within $\mathcal{D}(t)$ is checked to decide whether it will be included in the transmission schedule $x(t)$. For any $i \in \mathcal{D}(t)$ if no nodes in $\mathcal{C}(i)$ were active in the previous data slot, i.e., $\forall j \in \mathcal{C}(i), x(t-1) \neq j$, then i chooses to be *active* with probability p_i and *inactive* with probability $1-p_i$ in the current data slot. If at least one node in $\mathcal{C}(i)$ was active in the previous data slot, i.e., $\exists j \in \mathcal{C}(i), x(t-1) = j$, then i will be *inactive* in the current data slot. Any node $i \notin \mathcal{D}(t)$ maintains its state (active or inactive) from the previous data slot.

B. Relay Q-CSMA Algorithm (RQ-CSMA)

We propose RQ-CSMA algorithm to obtain a feasible throughput optimal schedule, when the channels to the destination are randomly varying over time. In RQ-CSMA, a time slot is divided into a contention and a transmission slot as in Q-CSMA. Unlike Q-CSMA, the purpose of the contention slot in RQ-CSMA is not only to generate a collision free schedule to be used in the transmission slot but also to infer the channel states of every node, i.e., $\mathbf{s}(t) = (s_0(t), s_1(t), \dots, s_N(t))$.

Let us denote by $\mathcal{D}^\theta(t)$, the decision schedule at time slot t with $\mathbf{s}(t) = \theta$, where $\theta = \{\theta_0, \theta_1, \dots, \theta_N\}$, and $\theta_i \in \{0, 1\}$ is a realization of randomly varying channel states. Note that in the relay network considered, the cardinality of set $\mathcal{D}^\theta(t)$ is always one, since only one node can be scheduled without conflicting with others. The network selects a decision schedule $\mathcal{D}^\theta(t)$ randomly with probability $\alpha(\mathcal{D}^\theta(t))$, where $\sum_{\mathcal{D}^\theta(t) \in \mathcal{M}^\theta} \alpha(\mathcal{D}^\theta(t)) = 1$, and \mathcal{M}^θ is the set of all possible feasible schedules of the network when $\mathbf{s}(t) = \theta$. Note that in the relay network considered, a schedule is feasible, if only one node is scheduled, and the scheduled node $i = 0, 1, \dots, N$ has its channel in ON state, i.e., $s_i(t) = 1$. Meanwhile, it is always feasible to schedule node 0, since links between node 0 and other nodes are always ON.

Unlike Q-CSMA, in RQ-CSMA, the transmission schedule is determined according to the channel states. Let $y^\theta(t)$ be the transmission schedule in the most recent data slot (before and

including time t) when $s(t) = \vartheta$. More specifically,

$$y^\vartheta(t) = \{x(\tau^*) : \tau^* = \max\{\tau \text{ s.t. } \tau \leq t \text{ and } s(\tau) = \vartheta\}\}. \quad (9)$$

y^ϑ is the transmission activity of the nodes in the last instance of time, before and including t , where the channel state vector realization was ϑ . Consider the following example to better illustrate the evolution of y^ϑ . Assume that only the channel of node 0 is time varying. Since there is only one time varying channel, there are only two variables defined by (9), i.e., $y^1(t)$ and $y^0(t)$. The sample path evolution of $y^1(t)$, $y^0(t)$ and $x(t)$ are given in Fig. 4. Suppose that at time $t = 8$, RQ-CSMA has chosen to schedule an *idle* slot. Note that since at $t = 8$, the channel state is 0, $y^0(8)$ updates its value to the value of $x(8)$ while $y^1(8)$ maintains its value from $t = 7$, since it is not allowed to change its value. Basically, when $s_0(t) = 0$, only $y^0(t)$ evolves, and $y^1(t)$ does not evolve. Note that in scheduling the nodes, if $s_0(t) = 1$ ($s_0(t) = 0$), RQ-CSMA uses $y^1(t)$ ($y^0(t)$).

At time slot t , based on the realization of the channel state vector $s(t) = \vartheta$ a decision schedule, $\mathcal{D}^\vartheta(t)$ is randomly determined. Then each node i searches for the corresponding y to the channel state vector $s(t) = \vartheta$, i.e., y^ϑ . If node $i \in \mathcal{D}^\vartheta(t)$, and $y^\vartheta(t-1) \neq j$ for all $j \neq i$, then node i chooses to be active with probability p_i^ϑ , and idle with probability $1 - p_i^\vartheta$. If $y^\vartheta(t-1) = j$ for any $j \neq i$, then i will be inactive in the current transmission slot. Any node i not selected in $\mathcal{D}^\vartheta(t)$ will maintain its previous state, i.e., it will be inactive if $y^\vartheta(t-1) \neq i$ and active otherwise. The conditions on activation probabilities p_i^ϑ will be specified in Section IV-C. Note that in Q-CSMA, transmission schedule is determined based on the previous transmission schedule $x(t-1)$, whereas in RQ-CSMA we use $y^\vartheta(t-1)$ for each ϑ to determine the transmission schedule $x(t)$. In Section IV-C, We will show that defining y^ϑ as such, and using it to schedule the transmissions, results in Markov chains with product-form stationary distributions. This is then used to prove throughput optimality of RQ-CSMA. The detailed description of RQ-CSMA is given in Algorithm 1.

Algorithm 1 RQ-CSMA

At each time slot, each node i performs the following procedure.

- 1: In *control* slot, randomly select a decision schedule $\mathcal{D}^\vartheta(t) \in \mathcal{M}^\vartheta$ with probability $\alpha(\mathcal{D}^\vartheta(t))$ (node i contends to be included in the decision schedule.),
 - 2: **if** $i \in \mathcal{D}^\vartheta(t)$ and $y^\vartheta(t-1) \neq j$ for all $j \neq i$ **then**
 - 3: $x(t) = i$ with probability p_i^ϑ
 - 4: $x(t) = \mathcal{J}$ with probability $1 - p_i^\vartheta$
 - 5: **else if** $i \in \mathcal{D}^\vartheta(t)$ and $y^\vartheta(t-1) = j$ for some $j \neq i$ **then**
 - 6: node i does not transmit.
 - 7: **else**
 - 8: node i transmits if $y^\vartheta(t-1) = i$, otherwise node i does not transmit
 - 9: In the *data* slot, use $x(t)$ as the transmission schedule
-

We illustrate the operation of RQ-CSMA with the following simple example.

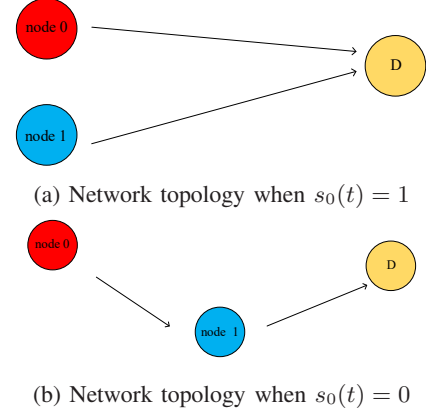


Fig. 3: Different network topologies associated with $s_0(t)$

Example 1: Suppose that the network consists of a source and a relay node with only node 0 having a time varying channel, i.e., $s(t) = s_0(t)$. Note that whenever $s_0(t) = 1$, the network consists of two nodes willing to transmit to a common destination as in Figure 3a. In this scenario, the possible decision schedules are $\mathcal{D}^1 = \{\mathcal{J}\}$ or $\{0\}$ or $\{1\}$, which translates into two nodes being idle, or direct transmission of node 0 to destination or transmission of node 1 to destination, respectively. Meanwhile, if $s_0(t) = 0$, the network consists of a node 0 which forwards its packet to the node 1 for eventual delivery to the destination as shown in Figure 3b. In this scenario, the possible decision schedules are $\mathcal{D}^0 = \{\mathcal{J}\}$ or $\{0\}$ or $\{1\}$, which translates into two nodes being idle, or node 0 relaying a packet to node 1 or transmission of node 1 to destination, respectively.

Since there is only one time varying channel, we need to define $y^1(t)$, the transmission schedule corresponding to the last instance τ^* that $s_0(\tau^*) = 1$ and $y^0(t)$, the transmission schedule corresponding to the last instance τ^* such that $s_0(\tau^*) = 0$ according to (9). Note that $y^1(t)$ is associated with Figure 3a and $y^0(t)$ is associated with Figure 3b.

First consider the case when $s_0(t) = 1$, $y^1(t-1) = \mathcal{J}$, i.e., in the most recent time slot, τ , when $s_0(\tau) = 1$, where $\tau < t$, both nodes did not transmit. Then,

- if the decision schedule in the current slot is $\mathcal{D}^1(t) = \{0\}$, then node 0 transmits to the destination with probability p_0^1 and remains idle with probability $1 - p_0^1$.
- if the decision schedule is $\mathcal{D}^1(t) = \{1\}$, then node 1 transmits to the destination with probability p_1^1 , and it remains idle with probability $1 - p_1^1$.

Now, consider the other possibility for the channel state of node 0, i.e., when $s_0(t) = 0$, we have $y^0(t-1) = 1$, which means that in the most recent time slot, τ , when $s_0(\tau) = 0$, node 0 was idle whereas node 1 transmitted. Then,

- if the decision schedule in the current slot is $\mathcal{D}^0(t) = \{0\}$, then node 0 does not transmit, because the last time when the node 0 to destination channel was OFF, node 1 was active.
- if the decision schedule in the current slot is $\mathcal{D}^0(t) = \{1\}$, then node 1 transmits to the destination with probability p_1^0 , and it remains idle with probability $1 - p_1^0$.

t	5	6	7	8	9	10	11	12
$s_0(t)$	1	0	0	0	0	1	1	0
$y^1(t)$	3	3	3	3	3	3	2	2
$y^0(t)$	0	0	0	3	2	2	2	3
$x(t)$	3	0	0	3	2	3	2	3

Fig. 4: The evolution of $y^1(t)$, $y^0(t)$ and $x(t)$. Numbers in black mean that the schedule is allowed to evolve, while the numbers in blue mean that it remains the same.

C. Contention Resolution Scheme

In this section, we propose a contention resolution scheme which satisfies two purposes: i) Randomly choosing a feasible schedule, and ii) Inferring the channel state vector $\mathbf{s}(t)$ at each node in a *distributed manner*. At the beginning of contention slot, the destination node broadcasts a probe packet. Upon successful detection of this packet, node i determines that its channel to the destination node is ON at slot t , and otherwise, it recognizes that its channel is OFF. We assume that the contention slot is divided into $N + 1 + W$ number of mini slots. Each node is registered to one of the first $N + 1$ mini slots, i.e., we assign i^{th} , $i = 0, \dots, N + 1$, mini slot to i^{th} node. At i^{th} mini slot, node i broadcasts a random signal, and all other nodes attempt to detect the random message. If they detect a message on i^{th} mini-slot, they conclude that $s_i(t) = 1$, otherwise $s_i(t) = 0$. Hence, at the end of $N + 1$ mini-slots, all nodes are aware of other nodes' channel states to the destination.

To randomly choose (i.e., with probability $\alpha(\mathcal{D}^\vartheta(t))$ for any $\vartheta \in \mathcal{S}$) a decision schedule $\mathcal{D}^\vartheta(t)$, node i (given that $s_i(t) = 1$) randomly selects a number T_i uniformly distributed in $[N + 1, N + 1 + W]$, and waits for T_i mini slots. If node i hears an INTENT message before $T_i + 1$, it will not be included in $\mathcal{D}^\vartheta(t)$. Otherwise, node i broadcasts an INTENT message at the beginning of the $(T_i + 1)^{th}$ control mini-slot. If there is no collision in $(T_i + 1)^{th}$ control mini-slot, node i will be included in $\mathcal{D}^\vartheta(t)$. Finally, if there is a collision, none of the nodes will be included in $\mathcal{D}^\vartheta(t)$ (i.e., $\mathcal{D}^\vartheta(t) = \{\mathcal{J}\}$).

Note that the contention resolution scheme requires all nodes to hear each other which may not be always feasible. This is similar to well-known *hidden node* problem in wireless networks which can be resolved by Request-To-Decide (RTD) and Clear-To-Decide (CTD) mechanism. The INTENT message in our original scheme is split into RTD and CTD pair transmitted and received, respectively, in two successive sub-mini slots. In the first sub-mini slot, node i , $i = 1, \dots, N$ sends RTD to the destination node. If the destination receives RTD without a collision (i.e., no other nodes are transmitting in the same sub-mini slot), then the destination replies with CTD to node i in the second sub-mini slot. If node i receives the CTD from destination without a collision, then node i is added to the decision schedule.

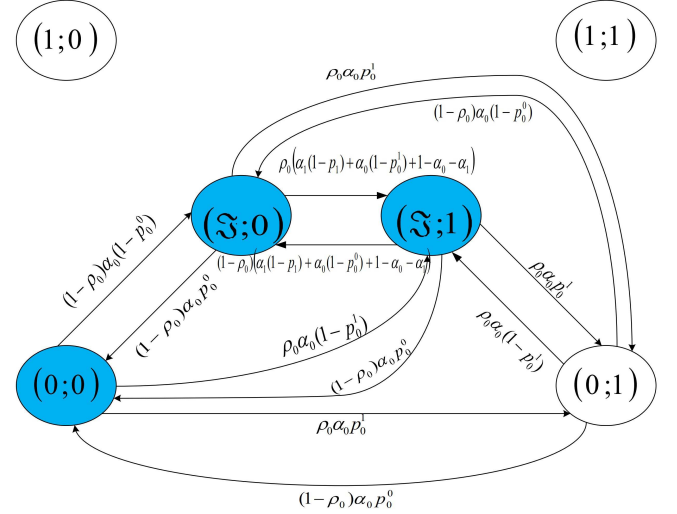


Fig. 5: DTMC with $(x; s_0)$ as states for Example 1.

TABLE I: SUMMARY OF NOTATION

Notation	Description
$s_i(t)$	Channel state between node i and the destination at time t ,
$A_i(t)$	Number of packets arriving to node i at time t ,
$Q_i(t)$	Size of the queue, storing the packets of node i at time t ,
$Q_{0i}(t)$	Relaying queue of node i at time t ,
$x(t)$	Scheduled node at time t ,
$\mathcal{D}(t)$	The decision schedule,
$\mathcal{D}^\vartheta(t)$	The decision schedule when $\mathbf{s}(t) = \vartheta$,
$y^\vartheta(t)$	Transmission schedule in the most recent data slot (before and including time t) when $\mathbf{s}(t) = \vartheta$,
$p_i^\vartheta(t)$	Activation probability of node i when $\mathbf{s}(t) = \vartheta$,
$\omega_x(t)$	Weight associated with scheduled node x at time t .

IV. THROUGHPUT OPTIMALITY OF RQ-CSMA

In this section, we first discuss why a direct application of Q-CSMA is not possible for our relay network. Then, we explain how we can modify the standard analysis of throughput optimality according to the structure of our proposed RQ-CSMA algorithm. Finally, we provide the proof of throughput optimality of RQ-CSMA when the channel states are determined correctly according to the contention algorithm discussed earlier. The key notations used in this paper are listed in Table I.

A. Drawback of Q-CSMA

Even though Q-CSMA is known to achieve a throughput-optimal schedule in a distributed fashion for a wireless network with non-time-varying links, it is not a suitable choice for many other network models. A particular case was studied in [49], where the authors investigated a random access scheme for scheduling nodes in a CRN. The authors in [49] demonstrated that Q-CSMA is not throughput optimal in this network model, since the Discrete Time Markov Chain (DTMC) with the transmission schedule $x(t)$ chosen as the state is not time-reversible, and thus, it does not have a product-form solution.

Similarly, Q-CSMA is not a suitable choice for our system model as well. Note that Q-CSMA is throughput optimal when the activation probabilities for each node are chosen to be a function of the node weights as defined in (6) and (7) of [29]. It can be argued that the stationary distribution of DTMC exists in steady state, if the activation probabilities change slowly over time. Unfortunately, in our system model, the node weights may vary abruptly depending on the channel state. Specifically, in *Example 1*, whenever $s_0(t) = 1$, the weight of node 0 is $f_0(Q_0(t))$ and it becomes $f_0(Q_0(t) - Q_{01}(t))$ otherwise. Consider the case where $Q_0(t)$ and $Q_{0i}(t)$ are both very large and equal. In this case, the activation probability of node 0 is approximately equal to 1 when $s_0(t) = 1$ and 0 otherwise. This results in a transition probability jumping from 1 to 0 from one slot to the next. This implies that the distribution of DTMC cannot be assumed to be at steady state at a given time³. Moreover, the DTMC in our system model is not time-reversible either.

B. Modifying the DTMC with channel state information

Next, we demonstrate that associating the channel state of the node 0 into the states cannot resolve this problem. For this purpose, we again consider the network in *Example 1*. Let $(x; s_0)$ be the states of the new DTMC as shown in Figure 5⁴. Consider the highlighted states in Figure 5, and let us check whether the DTMC is time reversible by examining the transitions from $(0; 0)$, $(\mathcal{I}; 0)$ to $(\mathcal{I}; 1)$ in the clockwise and counter-clockwise directions. Let α_i , $i = 0, 1$, be the probability that node i is included in the decision schedule. The product of clockwise transition probabilities is $\rho_0(1 - \rho_0)^2 \alpha_0^2 p_0^0 (1 - p_0^0) (\alpha_1(1 - p_1) + \alpha_0(1 - p_0^1) + 1 - \alpha_0 - \alpha_1)$ and the product of counter-clockwise transition probabilities is $\rho_0(1 - \rho_0)^2 \alpha_0^2 p_0^0 (1 - p_0^0) (\alpha_1(1 - p_1) + \alpha_0(1 - p_0^0) + 1 - \alpha_0 - \alpha_1)$. Since, $p_0^1 \neq p_0^0$, these two probabilities are not equal and therefore the DTMC is not time reversible by Kolmogorov's criterion [50].

Inspired by the approach of [49], we define a separate DTMC^ϑ for each of the different realizations of channel state vector $\vartheta \in \mathcal{S}(t)$ to mitigate this problem. Let $y^\vartheta \in \mathcal{M}^\vartheta$ be the states of DTMC^ϑ . Note that a transition from state y^ϑ to another state \hat{y}^ϑ in DTMC^ϑ only depends on the current state

³Henceforth, Proposition 2 in [29] which assumes that the DTMC is in steady state at any given time does not hold for our network model.

⁴For the sake of simplicity, we only show the transition probabilities of the node 0.

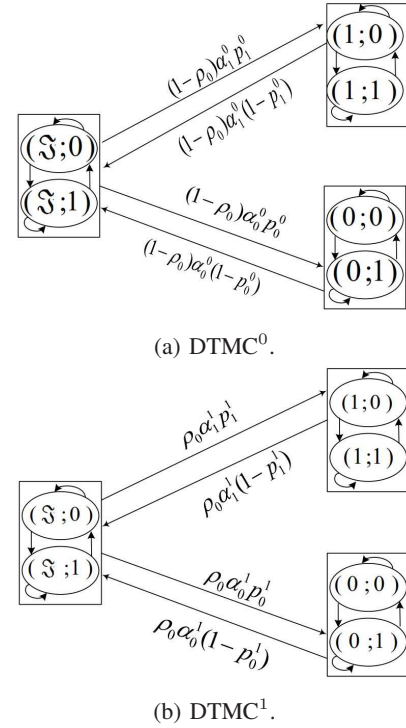


Fig. 6: Two different DTMCs associated with $s_0(t)$.

y^ϑ given the current channel state vector $\mathbf{s}(t)$ and the decision schedule. Thus, DTMC^ϑ is Markovian for all $\vartheta \in \mathcal{S}$.

Figure 6 shows two different DTMCs associated with each of two different states of s_0 for the network in *Example 1*. The transition probabilities follow directly from the definition of RQ-CSMA algorithm. Figure 6a represents DTMC^0 where a “Rectangle” corresponds to y^0 and an oval corresponds to $(y^0; s_0)$. For example, consider $(1; 0)$ in DTMC^0 ; being in this state means that the channel of the node 0 is OFF and only node 1 is transmitting, while $(1; 1)$ means that the channel of the node 0 is ON and the last time that channel was OFF prior to the current time slot, node 1 was transmitting. This follows from the definition of y^ϑ in (9). Similarly, Figure 6b represents DTMC^1 where a “Rectangle” corresponds to y^1 and an oval corresponds to $(y^1; s_0)$. Note that according to RQ-CSMA, any state, y^0 , (i.e., a rectangle) in DTMC^0 at any time t can make a transition into another state, \hat{y}^0 ($y^0 \neq \hat{y}^0$), if and only if $s_0(t) = 0$, and the state freezes otherwise. The same argument follows for DTMC^1 .

Now consider y^0 and y^1 as states for DTMC^0 and DTMC^1 , respectively⁵. The stationary distribution of DTMC^0 is $\pi(\mathcal{I}) = \frac{1}{Z^0}$, $\pi(1) = \frac{1}{Z^0} \frac{p_1^0}{1-p_1^0}$ and $\pi(0) = \frac{1}{Z^0} \frac{p_0^0}{1-p_0^0}$ where $Z^0 = 1 + \frac{p_0^0}{1-p_0^0} + \frac{p_1^0}{1-p_1^0}$. Similarly the stationary distribution of DTMC^1 is $\pi(\mathcal{I}) = \frac{1}{Z^1}$, $\pi(1) = \frac{1}{Z^1} \frac{p_1^1}{1-p_1^1}$ and $\pi(0) = \frac{1}{Z^1} \frac{p_0^1}{1-p_0^1}$ where $Z^1 = 1 + \frac{p_0^1}{1-p_0^1} + \frac{p_1^1}{1-p_1^1}$. Both DTMC^0 and DTMC^1 are time reversible and have product form stationary distributions.

⁵Note that Markov chains with $(y^0; s_0)$ and $(y^1; s_0)$ as states are not time reversible. It is easy to verify this observation from Fig.6, where the outgoing probabilities from $(y^0; 0)$ and $(y^0; 1)$ to $(\hat{y}^0; 0)$ are the same whereas the incoming probabilities from $(\hat{y}^0; 1)$ to $(y^0; s_0)$ do not exist if $y^0 \neq \hat{y}^0$.

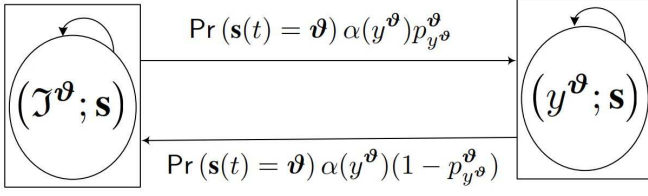


Fig. 7: Transition probabilities of states in DTMC^ϑ for any arbitrary $y^\vartheta \in \mathcal{M}^\vartheta$.

Moreover, at any given time only one of the DTMCs evolve. Also, the transition probabilities of DTMC^0 only depend on p_0^0 and p_1^0 , whereas the transition probabilities in DTMC^1 only depends on p_0^1 and p_1^1 . Hence, we can assume that at any time the DTMCs are at steady state.

C. Proving the Throughput Optimality of RQ-CSMA

In the following, we consider the general network topology with a source and N relay nodes, with all channels to the destination being time varying ON-OFF channels, whereas the channels from node 0 to all other nodes are always ON. We show that DTMC^ϑ for all $\vartheta \in \mathcal{S}$ is time reversible with a product-form stationary distribution. Theorem 1 gives the product-form of the stationary distribution, and finally, Theorem 3 claims the throughput optimality of RQ-CSMA.

Theorem 1. *DTMC $^\vartheta$ for any $\vartheta \in \mathcal{S}$ is reversible and it has the following product-form stationary distribution:*

$$\begin{aligned} \pi(y^\vartheta) &= \frac{1}{Z^\vartheta} \frac{p_{y^\vartheta}^\vartheta}{1 - p_{y^\vartheta}^\vartheta}, \\ \pi(\mathcal{J}^\vartheta) &= \frac{1}{Z^\vartheta}, \end{aligned} \quad (10)$$

$$\text{where } Z^\vartheta = 1 + \sum_{y^\vartheta \in \mathcal{M}^\vartheta} \frac{p_{y^\vartheta}^\vartheta}{1 - p_{y^\vartheta}^\vartheta}. \quad (11)$$

Proof. Consider the state transitions of DTMC^ϑ as given in Fig. 7. For any $y^\vartheta \in \mathcal{M}^\vartheta$ we can write the following detailed balance equation which follows from RQ-CSMA:

$$\begin{aligned} \pi(\mathcal{J}^\vartheta) \Pr(s(t) = \vartheta) \alpha(y^\vartheta) p_{y^\vartheta}^\vartheta \\ = \pi(y^\vartheta) \Pr(s(t) = \vartheta) \alpha(y^\vartheta) (1 - p_{y^\vartheta}^\vartheta), \end{aligned} \quad (12)$$

which simplifies to:

$$\pi(y^\vartheta) = \pi(\mathcal{J}^\vartheta) \frac{p_{y^\vartheta}^\vartheta}{1 - p_{y^\vartheta}^\vartheta}, \quad (13)$$

Next we have,

$$\begin{aligned} \pi(\mathcal{J}^\vartheta) + \sum_{y^\vartheta \in \mathcal{M}^\vartheta} \pi(y^\vartheta) \\ = \pi(\mathcal{J}^\vartheta) + \sum_{y^\vartheta \in \mathcal{M}^\vartheta} \pi(\mathcal{J}^\vartheta) \frac{p_{y^\vartheta}^\vartheta}{1 - p_{y^\vartheta}^\vartheta} = 1. \end{aligned} \quad (14)$$

If we define $Z^\vartheta = 1 + \sum_{y^\vartheta \in \mathcal{M}^\vartheta} \frac{p_{y^\vartheta}^\vartheta}{1 - p_{y^\vartheta}^\vartheta}$, we have:

$$\pi(\mathcal{J}^\vartheta) = \frac{1}{Z^\vartheta} \quad (15)$$

Substituting (15) in (13) gives the stationary distribution of DTMC^ϑ . \square

The product-form stationary distribution of DTMC^ϑ , suggests that we can use the results established in [13] to prove the throughput optimality of the algorithm. Let $x(t)$ be the schedule at time slot t . For the sake of simplicity, we drop t and denote the weight associated with the scheduled node $x(t)$ by $\omega_x(t)$. Similarly, $\omega_{y^\vartheta}(t)$ is the weight associated with the $y^\vartheta(t)$.

Theorem 2. [13] *Let $\omega^*(t) := \max_{x \in \mathcal{M}(t)} \omega_x(t)$, where $\mathcal{M}(t)$ is the set of all feasible schedules at time t . For a scheduling algorithm, for any $0 < \epsilon, \delta < 1$, if there exists $\beta > 0$ such that if $\omega^*(t) > \beta$, the scheduling algorithm chooses a schedule $x(t) \in \mathcal{M}(t)$ that satisfies*

$$\Pr\{\omega_x(t) \geq (1 - \epsilon)\omega^*(t)\} \geq 1 - \delta \quad (16)$$

where $\omega_x(t)$ is a function of the lengths of queues defined in (6) and (7). Then the scheduling algorithm is throughput optimal.

If we choose $p_{y^\vartheta}^\vartheta = \frac{\exp(\omega_{y^\vartheta}(t))}{1 + \exp(\omega_{y^\vartheta}(t))}$, then the stationary distribution of DTMC^ϑ becomes:

$$\begin{aligned} \pi(y^\vartheta) &= \frac{\exp(\omega_{y^\vartheta}(t))}{Z^\vartheta}, \\ \pi(\mathcal{J}^\vartheta) &= \frac{1}{Z^\vartheta} \end{aligned} \quad (17)$$

$$Z^\vartheta = 1 + \sum_{y^\vartheta \in \mathcal{M}^\vartheta} \exp(\omega_{y^\vartheta}(t)) \quad (18)$$

By choosing f_i 's in (6) and (7) wisely, $p_{y^\vartheta}^\vartheta$ changes slowly over time and we can assume that DTMC^ϑ is in steady-state in every time slot (time scale separation) [29]. In the following we show that RQ-CSMA is throughput optimal by showing that it is close enough to another throughput optimal algorithm (i.e., MWS).

Theorem 3. *For any $\vartheta \in \mathcal{S}$, suppose that $\bigcup_{\mathcal{D}^\vartheta \in \mathcal{M}^\vartheta} \mathcal{D}^\vartheta = \mathcal{M}^\vartheta$. Let $p_{y^\vartheta}^\vartheta = \frac{\exp(\omega_{y^\vartheta}(t))}{1 + \exp(\omega_{y^\vartheta}(t))}$, $\forall y^\vartheta \in \mathcal{M}^\vartheta$ when $s(t) = \vartheta$. Then RQ-CSMA is throughput-optimal.*

Proof. We define $\omega^\vartheta(t) := \max_{x \in \mathcal{M}^\vartheta(t)} \omega_x(t)$. Next define the following states:

$$\chi^\vartheta = \{(y^\vartheta; \vartheta) | s.t., \omega_{y^\vartheta}(t) < (1 - \epsilon)\omega^\vartheta(t)\}, \quad (19)$$

$$\Psi^\vartheta = \bigcup_{\vartheta \in \mathcal{S}} \chi^\vartheta \quad (20)$$

$$\varphi^\vartheta = \{y^\vartheta | s.t., \omega_{y^\vartheta}(t) < (1 - \epsilon)\omega^\vartheta(t)\}. \quad (21)$$

Note that $\pi(\bigcup_{\vartheta \in \mathcal{S}} (y^\vartheta; \vartheta)) = \pi(y^\vartheta)$ so $\pi(\Psi^\vartheta) = \pi(\varphi^\vartheta)$.

We now calculate the probability of a state in set χ^ϑ :

$$\begin{aligned}
\pi(\chi^\vartheta) &< \pi(\Psi^\vartheta) = \pi(\varphi^\vartheta) \\
&= \sum_{y^\vartheta \in \varphi^\vartheta} \pi(y^\vartheta) \\
&= \sum_{y^\vartheta \in \varphi^\vartheta} \frac{\exp(\omega_{y^\vartheta}(t))}{Z^\vartheta} \\
&\leq \frac{|\chi^\vartheta| (1 - \epsilon) \omega^\vartheta(t)}{Z^\vartheta} \\
&< \frac{N + 2}{\exp(\epsilon \omega^\vartheta(t))},
\end{aligned} \tag{22}$$

where (22) is true because $|\chi^\vartheta| \leq |\mathcal{M}^\vartheta| \leq N + 2$, and

$$Z^\vartheta > \exp\left(\max_{y^\vartheta \in \mathcal{M}^\vartheta} \omega_{y^\vartheta}\right) > \exp(\omega^\vartheta(t)). \tag{23}$$

Therefore, if

$$\omega^\vartheta(t) > \frac{1}{\epsilon} \left(\log(N + 2) + \log \frac{1}{\delta^\vartheta} \right), \tag{24}$$

then $\pi(\chi^\vartheta) < \delta^\vartheta$. Then we have the following results:

$$\begin{aligned}
&\Pr\{\omega_{y^\vartheta}(t) \geq (1 - \epsilon) \omega^\vartheta(t) | \mathbf{s}(t) = \vartheta\} \\
&= 1 - \Pr\{\omega_{y^\vartheta}(t) < (1 - \epsilon) \omega^\vartheta(t) | \mathbf{s}(t) = \vartheta\} \\
&= 1 - \frac{\frac{N+2}{\epsilon \omega^\vartheta(t)}}{\Pr(\mathbf{s}(t) = \vartheta)} := 1 - \delta^\vartheta.
\end{aligned} \tag{25}$$

Let x be the node scheduled in slot t by RQ-CSMA. Using the total probability law we have:

$$\begin{aligned}
&\Pr\{\omega_x(t) \geq (1 - \epsilon) \omega^*(t)\} \\
&= \sum_{\vartheta \in \mathcal{S}} \{ \Pr\{\omega_{y^\vartheta}(t) \geq (1 - \epsilon) \omega^\vartheta(t) | \mathbf{s}(t) = \vartheta\} \\
&\quad \times \Pr(\mathbf{s}(t) = \vartheta) \}
\end{aligned} \tag{26}$$

$$\geq (1 - \delta) \sum_{\vartheta \in \mathcal{S}} \Pr(\mathbf{s}(t) = \vartheta) = 1 - \delta, \tag{27}$$

where $\delta = \max_{\vartheta \in \mathcal{S}} \delta^\vartheta$. Note that (26) holds because $x = y^\vartheta$ and $\omega^*(t) = \omega^\vartheta(t)$ whenever $\mathbf{s}(t) = \vartheta$. Hence, Algorithm 1 satisfies the condition of Theorem 2, and thus, it is throughput optimal. \square

D. On the complexity of the RQ-CSMA

At time slot t , each node i first needs to find a decision schedule, and the channel state vector $\mathbf{s}(t)$. Note that this requires $N+1+W$ mini slots which results in $O(N)$ complexity. Then in RQ-CSMA each node needs to find the appropriate DTMC $^\vartheta$ scheduled for evolution which means that each node needs to find a single element from a set with size 2^{N+1} . It is well known that the simple binary search, has logarithmic complexity [51]. Consequently, the total complexity of RQ-CSMA is $O(\log(2^{N+1})) + O(N) = O(N)$.

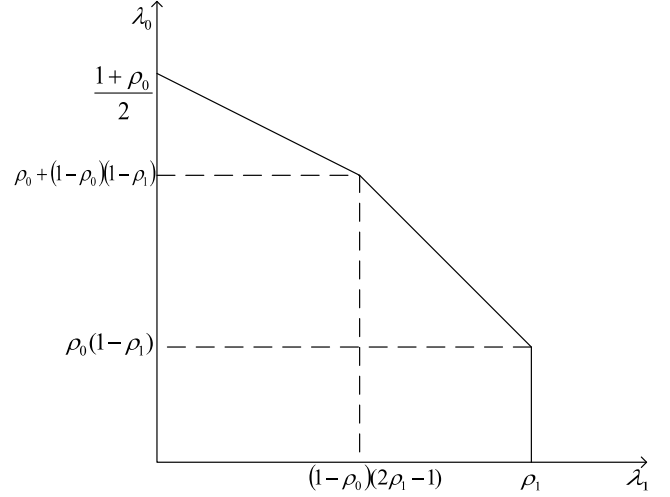


Fig. 8: Achievable rate region of the two node relay network with $\rho_1 \geq 0.5$.

V. ACHIEVABLE RATE REGION OF THE RELAY NETWORK

In order to validate the throughput optimality of RQ-CSMA, we focus on a network consisting of a source and a relay nodes. This helps us visualize the geometric representation of the achievable rate region and simplify the performance comparison of RQ-CSMA and Q-CSMA. Lemma 1 establishes the achievable rate region for two nodes with node 0 having arrival rate of λ_0 and probability of ON channel state, ρ_0 and node 1 having arrival rate of λ_1 and probability of ON channel state, ρ_1 :

Lemma 1. For two nodes, the achievable rate region of the relay network, when $\rho_1 < 0.5$, is

$$\Lambda^c = \{\lambda | \lambda_1 < \rho_1, \lambda_0 + \lambda_1 < \rho_0 + \rho_1(1 - \rho_0)\} \tag{28}$$

and for $\rho_1 \geq 0.5$ is

$$\begin{aligned}
\Lambda^c = \{ \lambda | &\lambda_1 < (1 - \rho_0)(2\rho_1 - 1), 2\lambda_0 + \lambda_1 < 1 + \rho_0; \\
&(1 - \rho_0)(2\rho_1 - 1) \leq \lambda_1 < \rho_1, \lambda_0 + \lambda_1 < \rho_0 + (1 - \rho_0)\rho_1 \}
\end{aligned} \tag{29}$$

Proof. The proof is given in Appendix. \square

The achievable rate region of the relay network, Λ^c , for the case of two nodes with $\rho_1 \geq 0.5$ and with $\rho_1 < 0.5$ is depicted in Figure 8 and 9, respectively. Note that in case of no relay service, node 0 can only achieve a rate of ρ_0 , whereas when relaying is enabled, it can achieve an additional rate of $\frac{1-\rho_0}{2}$ for $\rho_2 \geq 0.5$ and $\rho_1(1 - \rho_0)$ for $\rho_1 < 0.5$.

VI. NUMERICAL RESULTS

In this section, we numerically validate the throughput optimality of RQ-CSMA and compare its achievable rate region with those of Q-CSMA and a simple algorithm utilizing uniform random back-off [29], [49]. The uniform back-off (UB) algorithm is defined in Algorithm 2. Note that Q-CSMA is not designed for time varying channels so in Q-CSMA if a node is chosen to be activated when its channel is at OFF state, it remains idle in that slot. Also, note that we choose

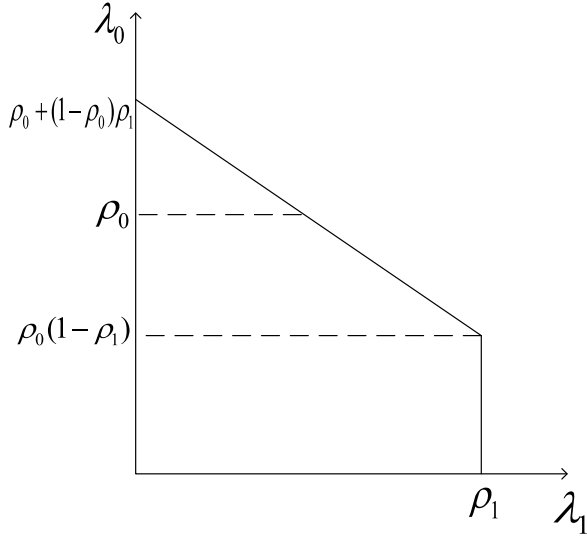


Fig. 9: Achievable rate region of the two node relay network with $\rho_1 < 0.5$.

f_i 's in (6) and (7) to be of the form $f(x) = \log(\beta x)$ with $\beta = 0.1$.

Note that we do not employ the RTD/CTD mechanism in the simulations, since it is required to be implemented by other policies as well to solve the hidden node problem, and its overhead will be similar in all. Hence, in the following we assume that any pairs of nodes are able to hear each other.

Algorithm 2 Uniform Backoff (UB)

At each time slot,

- 1: Node i selects a random back-off time $T_i = \text{Uniform}[1, W]$ and waits for T_i control mini slots.
 - 2: If node i hears an INTENT message from any other node before the $(T_i + 1)$ -th control mini-slot, it will not be included in the transmission schedule and will not transmit an INTENT message.
 - 3: If node i does not hear an INTENT message from any other node before the $(T_i + 1)$ -th control mini-slot, it will send an RESV message to all other nodes at the beginning of $(T_i + 1)$ -th control mini-slot.
 - If there is a collision, node i will not be included in transmission schedule, $x(t)$.
 - If there is no collision, node i will be included in the transmission schedule, $x(t)$.
 - 4: If $x(t) = i$, then node i will transmit. (Links with empty queues will keep silent in this time slot.)
-

A. Validating Throughput Optimality by Simulations

We consider a network consisting of two nodes. We evaluate the performance of RQ-CSMA according to the average queue lengths of all queues in the network, i.e., $Q_{avg} = \frac{1}{t} \sum_{\tau=0}^t (Q_0(\tau) + Q_1(\tau) + Q_{01}(\tau))$. To calculate Q_{avg} , we run the simulation for $t = 10^4$ slots, and over 10 different sample paths. The channel statistics are chosen as $\rho_0 = 0.4$, $\rho_1 = 0.7$. The arrival rates, (λ_1, λ_0) , are taken in the region

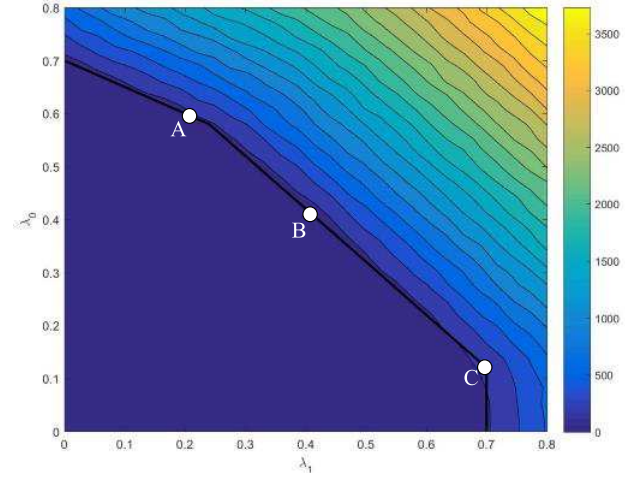


Fig. 10: Contour plot of Q_{avg} versus λ_0 and λ_1 evaluated by RQ-CSMA.

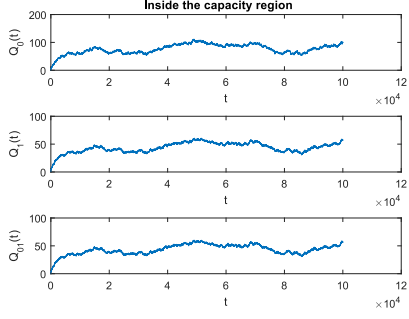
$[0, \rho_1 + 0.1] \times [0, \frac{1+\rho_0}{2} + 0.1]$ where we evaluate Q_{avg} for each point in this region. Figure 10 shows the contour graph of Q_{avg} versus λ_0 and λ_1 , where the solid and thick black line is the achievable rate region of the relay network. We observe that the network under RQ-CSMA exhibits unstable behavior, shown by the increase in the average queue lengths (lighter colored contours) as we cross the boundary of the stability region.

Next, to further investigate the stability region of the RQ-CSMA, we plot the time evolution of the queues at certain points in the achievable rate region. We choose points $A = (0.6, 0.2)$, $B = (0.4, 0.4)$ and $C = (0.12, 0.7)$ on the boundary of the achievable rate region as depicted in Figure 10. Then, we evaluate the time evolution of the queues at $A - \epsilon$, $B - \epsilon$ and $C - \epsilon$, i.e., the points are inside the achievable rate region and they are sufficiently close to the boundary where $\epsilon = 0.01$. Also we evaluate the time evolution of the queues for $A + \epsilon$, $B + \epsilon$ and $C + \epsilon$, i.e., the points are outside the achievable rate region and they are sufficiently close to the boundary. Figures 11a, 12a and 13a show that whenever we take a point inside the achievable rate region close to the boundary, RQ-CSMA is able to stabilize the queues as we expect from the throughput optimality of RQ-CSMA.

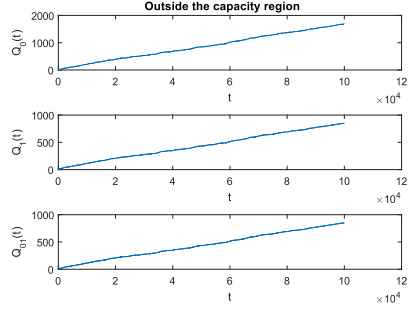
We repeat the same procedure for Q-CSMA and UB algorithms. Figure 14 and 16 show the contour graphs of Q_{avg} versus λ_0 and λ_1 under Q-CSMA and UB, respectively. We observe that the network under both Q-CSMA and UB exhibit unstable behavior, shown by the increase in the average queue lengths, even when the arrivals are far away from the boundaries of the stability region. To investigate further, we plot the time evolution of the queues under Q-CSMA and UB at $A - \epsilon$. The results are depicted in Figure 15 and 17. It can be seen that Q-CSMA and UB are unable to stabilize the queues for arrival rates stabilized by RQ-CSMA.

B. The effect of the number of relay nodes

Next, we numerically compare the performance of RQ-CSMA with UB, MWS and Q-CSMA algorithms when there

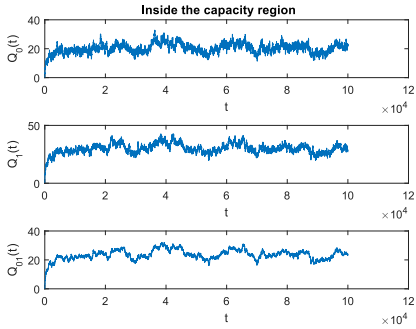


(a) Time evolution of queues at $A - \epsilon = (0.59, 0.19)$.

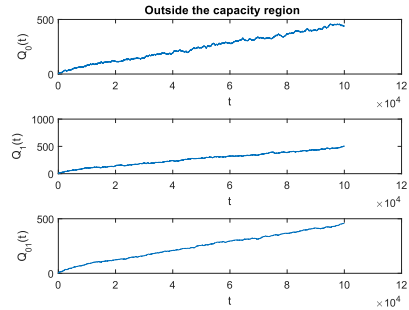


(b) Time evolution of queues at $A + \epsilon = (0.61, 0.21)$.

Fig. 11: Time evolution of queues for $A - \epsilon$ and $A + \epsilon$ under RQ-CSMA.

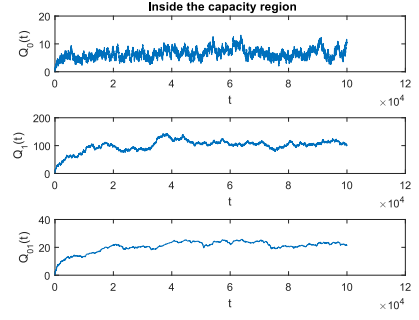


(a) Time evolution of queues at $B - \epsilon = (0.39, 0.41)$.

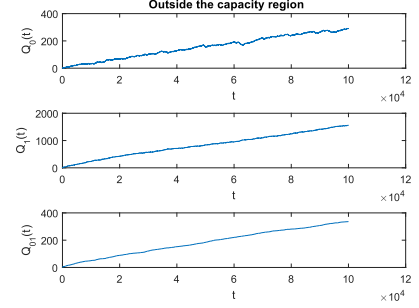


(b) Time evolution of queues at $B + \epsilon = (0.41, 0.43)$.

Fig. 12: Time evolution of queues for $B - \epsilon$ and $B + \epsilon$ under RQ-CSMA.



(a) Time evolution of queues at $C - \epsilon = (0.11, 0.69)$.



(b) Time evolution of queues at $C + \epsilon = (0.13, 0.71)$.

Fig. 13: Time evolution of queues for $C - \epsilon$ and $C + \epsilon$ under RQ-CSMA.

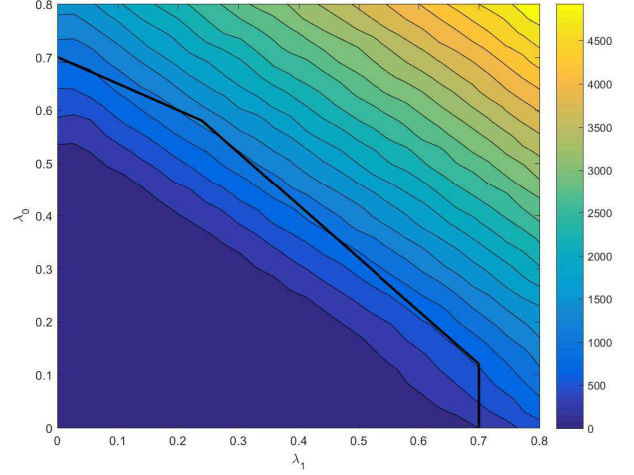


Fig. 14: Contour plot of Q_{avg} versus λ_0 and λ_1 evaluated by Q-CSMA.

is more than one relay node. For MWS algorithm, at each time slot, we choose an action for each node which maximizes the weights as defined in (6) and (7). We evaluated the average queue lengths of all nodes for a given arrival rate vector as our performance measure and illustrate 90% confidence intervals.

We consider a network with a source and three relay nodes. We choose $\rho_0 = 0.4$, $\rho_1 = 0.7$, $\rho_2 = 0.8$ and $\rho_3 = 0.7$ for the channel state statistics.

We compare the performance of the algorithms in terms of average queue lengths of all nodes. We begin with an arrival

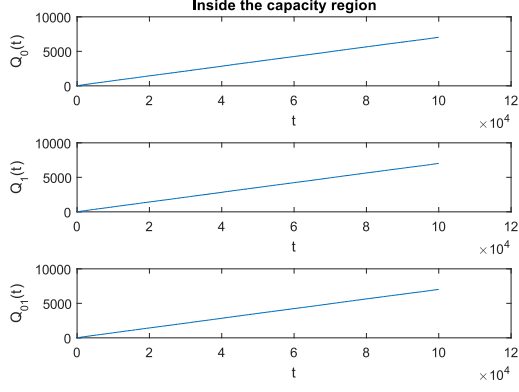


Fig. 15: Time evolution of queues for $A - \epsilon = (0.59, 0.19)$ under Q-CSMA.

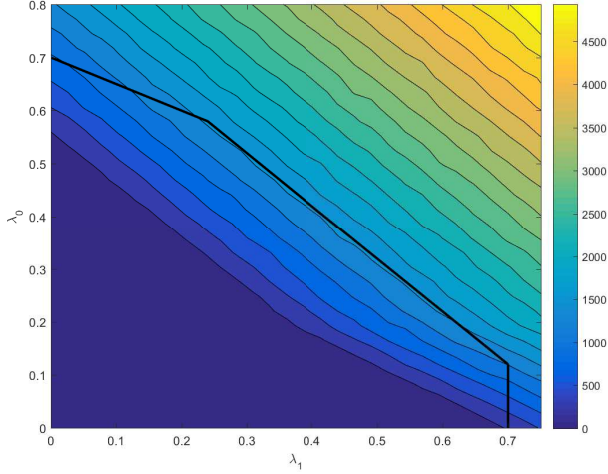


Fig. 16: Contour plot of Q_{avg} versus λ_0 and λ_1 evaluated by UB.

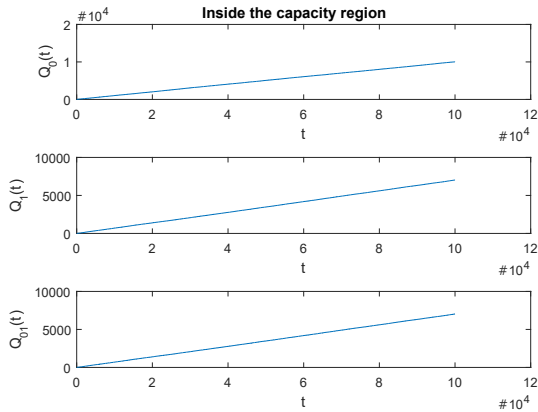


Fig. 17: Time evolution of queues for $A - \epsilon = (0.59, 0.19)$ under UB.

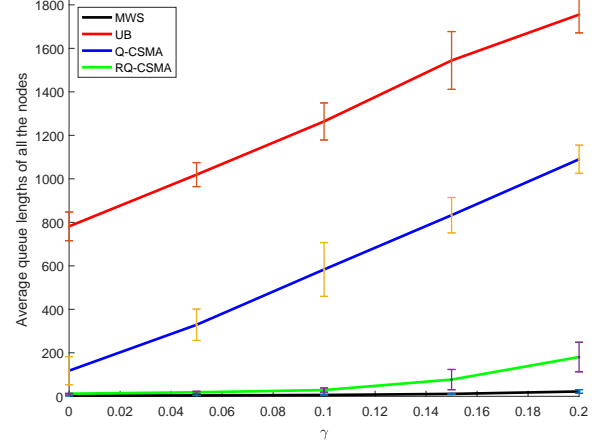


Fig. 18: The evolution of the average of the sum of queue lengths vs. γ over 10 random simulation runs and for an arrival rate vector $\lambda = (0.4 + \gamma, 0.05, 0.05, 0.05)$. The vertical error bars represent the 90% confidence intervals.

rate vector $\lambda = (0.4, 0.05, 0.05, 0.05)$ and linearly increase arrival rate of node 0 from 0.4 to $0.4 + \gamma$. For each algorithm, for a fixed γ we run 10 independent simulations and take the average. The average queue lengths of nodes with respect to γ is shown in Figure 18, where the vertical bars represent the 90% confidence interval. It can be seen that RQ-CSMA outperforms UB which does not take into account the queue length information and also Q-CSMA which does not take channel state information into account.

Although the average queue length information gives us an idea about the performance of these algorithms, it does not say much about the stability of the queues. Hence in the following, we compare the algorithms in terms of their queue length evolution with respect to time for a given arrival rate vector.

It can be seen from Figure 18 that for $\gamma = 0.2$ (i.e., $\lambda_0 = 0.6$) there is an increase in the average queue lengths of all nodes in all algorithms suggesting that the network is operating at a point close to the boundary of its achievable rate region. We pick the arrival rate vector as $\lambda = (0.6, 0.05, 0.05, 0.05)$ where the increase in average queue lengths of nodes is apparent in MWS algorithm according to Figure 18, i.e., λ is close to the boundary of the stability region. We again run each algorithm for arrival rate vector λ for 10 random seeds and then we take their average. The evolution of the queue lengths of nodes (where the sizes of all queues are summed) versus time is given in Figure 19, where the vertical bars represent the 90% confidence interval. It can be seen that for the desired arrival rate vector, MWS and RQ-CSMA can stabilize the queues whereas in UB and Q-CSMA the queue sizes tend to increase as the time evolves suggesting the instability of the queues.

VII. CONCLUSION

In this paper, we addressed the problem of scheduling in wireless relay networks with a source and multiple relay nodes

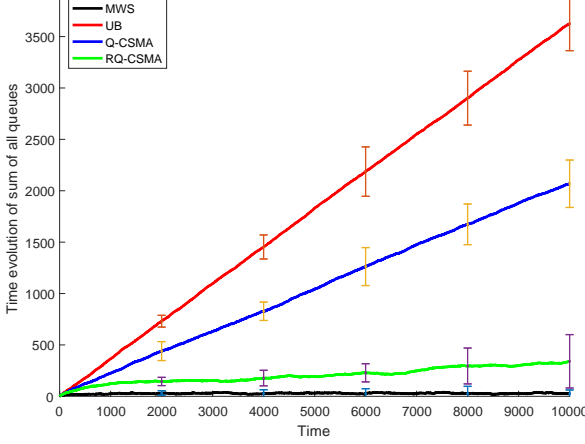


Fig. 19: The evolution of the average of the sum of queue lengths vs. time over 10 random simulation runs and for an arrival rate vector $\lambda = (0.6, 0.05, 0.05, 0.05)$. The vertical error bars represent the 90% confidence intervals.

where all nodes transmit to a common destination node over independent ON-OFF channels. In this relay network, N relay nodes help a single node with bad channel quality by relaying its packets. The scheduling is based on a variant of well known Q-CSMA algorithm, and we prove the throughput optimality of the developed algorithm. We proposed a new contention resolution algorithm, which is used by the nodes to infer which nodes have ON channels; an information required by the scheduling algorithm for achieving throughput optimality. RQ-CSMA is different from Q-CSMA in the sense that it uses channel state information to schedule the nodes and it actually benefits from time variability of the channel states. The performance of the RQ-CSMA, Q-CSMA, MWS and a simple random back-off based algorithms are compared through simulations to demonstrate the efficacy of RQ-CSMA.

As a future work, we will consider the extension of RQ-CSMA to multi-hop and multi source nodes relay networks. Also note that the throughput optimality of RQ-CSMA depends on the assumption that nodes are able to perfectly infer the channel state vector in the contention slot. The investigation of scheduling algorithms with partial and incomplete channel state information is left as future work.

PROOF OF LEMMA 1

To simplify the proof, we assume that node 1 has a single queue and stores the received packets from node 0 in its own queue. Note that this does not change the stability region of the network. Define $\mu_0(t)$ and $\mu_1(t)$ as the instantaneous service rates of node 0 and node 1, respectively, and $\mu_{01}(t)$ as the instantaneous service rate of node 0 if it relays its packets to node 1. Note that $\mu_0(t)$, $\mu_1(t)$ and $\mu_{01}(t)$ take binary values 0 or 1 and satisfy the following inequality $\mu_0(t) + \mu_1(t) + \mu_{01}(t) \leq 1$. Then, the evolution of the queues are given as:

$$\begin{aligned} Q_0(t+1) &= \max[Q_0(t) - \mu_0(t) - \mu_{01}(t), 0] + A_0(t) \\ Q_1(t+1) &= \max[Q_1(t) - \mu_1(t), 0] + A_1(t) + \mu_{01}(t) \end{aligned} \quad (30)$$

It is well known that when MWS is applied the resulting service rates are throughput optimal. Thus, the expected service rates can be calculated as follow:

$$\begin{aligned} \mathbb{E}(\mu_0(t)|Q_0(t), Q_1(t)) &= \rho_0(1 - \rho_1) + \rho_0\rho_1\mathbb{1}_{\{Q_0(t) \geq Q_1(t)\}} \\ \mathbb{E}(\mu_{01}(t)|Q_0(t), Q_1(t)) &= (1 - \rho_0)(1 - \rho_1)\mathbb{1}_{\{Q_0(t) - Q_1(t) > 0\}} \\ &\quad + (1 - \rho_0)\rho_1\mathbb{1}_{\{Q_0(t) - Q_1(t) \leq 0\}} \\ \mathbb{E}(\mu_1(t)|Q_0(t), Q_1(t)) &= (1 - \rho_0)\rho_1\mathbb{1}_{\{Q_0(t) - Q_1(t) \leq 0\}} \\ &\quad + \rho_0\rho_1\mathbb{1}_{\{Q_0(t) < Q_1(t)\}} \end{aligned} \quad (31)$$

Denote $\mathbf{Q}(t) = (Q_0(t), Q_1(t))$. The drift in the Lyapunov function, $L(\mathbf{Q}(t)) = \frac{1}{2}(Q_0^2(t) + Q_1^2(t))$, can be written as:

$$\begin{aligned} L(\mathbf{Q}(t+1)) - L(\mathbf{Q}(t)) &= \frac{1}{2} \sum_{i=0}^1 (Q_i^2(t+1) - Q_i^2(t)) \\ &\leq \frac{(\mu_0(t) + \mu_{01}(t))^2 + A_0^2(t)}{2} \\ &\quad + Q_0(t)(A_0(t) - \mu_0(t) - \mu_{01}(t)) \\ &\quad + \frac{\mu_1^2(t) + (A_1(t) + \mu_{01}(t))^2}{2} \\ &\quad + Q_1(t)(A_1(t) + \mu_{01}(t) - \mu_1(t)) \end{aligned} \quad (32)$$

Now define B as a bound on the summation of first and third terms above as follows:

$$\frac{(\mu_0(t) + \mu_{01}(t))^2 + A_0^2(t)}{2} + \frac{\mu_1^2(t) + (A_1(t) + \mu_{01}(t))^2}{2} \leq B \quad (33)$$

Now let us assume that $\rho_1 < 0.5$ so that the achievable rate region is as in (28). First we focus on $\rho_0 \leq \lambda_0 < \rho_0 + (1 - \rho_0)\rho_1$ and $0 \leq \lambda_1 \leq \rho_1(1 - \rho_0)$. For small positive value of ϵ we have $\lambda_0 + \lambda_1 + \epsilon = \rho_0 + (1 - \rho_0)\rho_1$. We write the conditional Lyapunov drift as follow:

$$\begin{aligned} L(\mathbf{Q}(t+1)|Q_0(t) < Q_1(t)) - L(\mathbf{Q}(t)|Q_0(t) < Q_1(t)) &\leq B \\ &\quad + (\lambda_0 - \rho_0(1 - \rho_1))Q_0(t) + (\lambda_1 - \rho_1)Q_1(t) \\ &\leq B + (\lambda_0 - \rho_0(1 - \rho_1))Q_0(t) + (\lambda_1 - \rho_1)Q_0(t) \\ &\leq B - \epsilon Q_0(t) \leq B - \epsilon_1 \sum_{i=0}^1 Q_i(t) \end{aligned} \quad (34)$$

With a similar approach we can show that:

$$\begin{aligned} L(\mathbf{Q}(t+1)|Q_1(t) \leq Q_0(t) \leq 2Q_1(t)) \\ - L(\mathbf{Q}(t)|Q_1(t) \leq Q_0(t) \leq 2Q_1(t)) \\ \leq B - \epsilon_2 \sum_{i=0}^1 Q_i(t) \end{aligned} \quad (35)$$

Expectation over the $\mathbf{Q}(t)$ yields:

$$\mathbb{E}(L(\mathbf{Q}(t+1)) - L(\mathbf{Q}(t))) < B - \epsilon \sum_{i=0}^1 \mathbb{E}(Q_i(t)) \quad (36)$$

The above holds for all $t \in \{0, 1, 2, \dots\}$ and with the same analysis as above holds for all points inside the achievable

rate region defined by (28). Summing over $t \in \{0, 1, 2, \dots\}$ for some integer $T > 0$ yields (by telescoping sums):

$$\mathbb{E}(L(\mathbf{Q}(T)) - L(\mathbf{Q}(0))) \leq BT - \epsilon \sum_{t=0}^{T-1} \sum_{i=0}^1 \mathbb{E}(Q_i(t)) \quad (37)$$

Rearranging the terms and taking the limit yields:

$$\lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=0}^{T-1} \sum_{i=0}^1 \mathbb{E}(Q_i(t)) \leq \frac{B}{\epsilon} \quad (38)$$

Thus, all queues are strongly stable. The proof of the Lemma for $\rho_1 \geq 0.5$ is straightforward and can be done with a similar analysis as above. For more information refer to [52].

REFERENCES

- [1] B. Zafar, S. Gharekhloo, and M. Haardt, "Analysis of multihop relaying networks: Communication between range-limited and cooperative nodes," *IEEE Vehicular Technology Magazine*, vol. 7, no. 3, pp. 40–47, Sept 2012.
- [2] J. Boyer, D. D. Falconer, and H. Yanikomeroglu, "Multihop diversity in wireless relaying channels," *IEEE Transactions on Communications*, vol. 52, no. 10, pp. 1820–1830, Oct 2004.
- [3] Q. Guan, F. R. Yu, S. Jiang, V. C. M. Leung, and H. Mehrvar, "Topology control in mobile ad hoc networks with cooperative communications," *IEEE Wireless Communications*, vol. 19, no. 2, pp. 74–79, April 2012.
- [4] A. Bendjaballah, H. E. Ghannudi, L. Clavier, N. Rolland, and P. A. Rolland, "Multihop wireless communications with non regenerative relays," in *2006 European Conference on Wireless Technology*, Sept 2006, pp. 189–192.
- [5] L. Tassioulas and A. Ephremides, "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks," *Automatic Control, IEEE Transactions on*, vol. 37, no. 12, pp. 1936–1948, Dec 1992.
- [6] E. M. Yeh and R. A. Berry, "Throughput optimal control of cooperative relay networks," *IEEE Transactions on Information Theory*, vol. 53, no. 10, pp. 3827–3833, Oct 2007.
- [7] M. J. Neely, "Optimal backpressure routing for wireless networks with multi-receiver diversity," in *2006 40th Annual Conference on Information Sciences and Systems*, March 2006, pp. 18–25.
- [8] A. K. Sadek, K. J. R. Liu, and A. Ephremides, "Cognitive multiple access via cooperation: Protocol design and performance analysis," *IEEE Transactions on Information Theory*, vol. 53, no. 10, pp. 3677–3696, Oct 2007.
- [9] N. Pappas, J. Jeon, A. Ephremides, and A. Traganitis, "Wireless network-level partial relay cooperation," in *ISIT*, 2012, pp. 1122–1126.
- [10] O. Simeone, Y. Bar-Ness, and U. Spagnolini, "Stable throughput of cognitive radios with and without relaying capability," *IEEE Transactions on Communications*, vol. 55, no. 12, pp. 2351–2360, Dec 2007.
- [11] I. Krikidis, J. N. Laneman, J. S. Thompson, and S. McLaughlin, "Protocol design and throughput analysis for multi-user cognitive cooperative systems," *IEEE Transactions on Wireless Communications*, vol. 8, no. 9, pp. 4740–4751, September 2009.
- [12] I. Krikidis, N. Devroye, and J. S. Thompson, "Stability analysis for cognitive radio with multi-access primary transmission," *IEEE Transactions on Wireless Communications*, vol. 9, no. 1, pp. 72–77, January 2010.
- [13] A. Eryilmaz, R. Srikant, and J. Perkins, "Stable scheduling policies for fading wireless channels," *Networking, IEEE/ACM Transactions on*, vol. 13, no. 2, pp. 411–424, April 2005.
- [14] R. Loynes, "The stability of a queue with non-independent inter-arrival and service times," *Proc. Camb. Philos. Soc.*, vol. 58, pp. 497–520, 1962.
- [15] R. Urgaonkar and M. J. Neely, "Opportunistic scheduling with reliability guarantees in cognitive radio networks," *IEEE Transactions on Mobile Computing*, vol. 8, no. 6, pp. 766–777, June 2009.
- [16] S. Moharir and S. Shakkottai, "Maxweight vs. backpressure: Routing and scheduling in multi-channel relay networks," in *INFOCOM, 2013 Proceedings IEEE*, April 2013, pp. 1537–1545.
- [17] M. Andrews, K. Kumaran, K. Ramanan, A. Stolyar, R. Vijayakumar, and P. Whiting, "Scheduling in a queueing system with asynchronously varying service rates," in *Probability in the Engineering and Informational Sciences*, pp. 191–217.
- [18] L. Georgiadis, M. J. Neely, and L. Tassioulas, "Resource allocation and cross-layer control in wireless networks," *Found. Trends Netw.*, vol. 1, no. 1, pp. 1–144, Apr. 2006. [Online]. Available: <http://dx.doi.org/10.1561/13000000001>
- [19] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. New York, NY, USA: W. H. Freeman & Co., 1990.
- [20] G. Sharma, N. Shroff, and R. Mazumdar, "Maximum weighted matching with interference constraints," in *Pervasive Computing and Communications Workshops, 2006. PerCom Workshops 2006. Fourth Annual IEEE International Conference on*, March 2006, pp. 5 pp.–74.
- [21] N. Alon, L. Babai, and A. Itai, "A fast and simple randomized parallel algorithm for the maximal independent set problem," *Journal of Algorithms*, vol. 7, no. 4, pp. 567–583, 1986.
- [22] X. Wu, R. Srikant, and J. Perkins, "Scheduling efficiency of distributed greedy scheduling algorithms in wireless networks," *Mobile Computing, IEEE Transactions on*, vol. 6, no. 6, pp. 595–605, June 2007.
- [23] M. Halldrsson and J. Radhakrishnan, "Greedy is good: Approximating independent sets in sparse and bounded-degree graphs," *Algorithmica*, vol. 18, no. 1, pp. 145–163, 1997.
- [24] B. Birand, M. Chudnovsky, B. Ries, P. Seymour, G. Zussman, and Y. Zwols, "Analyzing the performance of greedy maximal scheduling via local pooling and graph theory," *Networking, IEEE/ACM Transactions on*, vol. 20, no. 1, pp. 163–176, Feb 2012.
- [25] X. Lin and N. Shroff, "The impact of imperfect scheduling on cross-layer rate control in wireless networks," in *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, vol. 3, March 2005, pp. 1804–1814 vol. 3.
- [26] A. Gupta, X. Lin, and R. Srikant, "Low-complexity distributed scheduling algorithms for wireless networks," *Networking, IEEE/ACM Transactions on*, vol. 17, no. 6, pp. 1846–1859, Dec 2009.
- [27] C. Joo and N. Shroff, "Performance of random access scheduling schemes in multi-hop wireless networks," in *Signals, Systems and Computers, 2006. ACSSC '06. Fortieth Asilomar Conference on*, Oct 2006, pp. 1937–1941.
- [28] L. Jiang and J. Walrand, "A distributed csma algorithm for throughput and utility maximization in wireless networks," Tech. Rep., 2009.
- [29] J. Ni, B. Tan, and R. Srikant, "Q-csma: Queue-length-based csma/ca algorithms for achieving maximum throughput and low delay in wireless networks," *Networking, IEEE/ACM Transactions on*, vol. 20, no. 3, pp. 825–836, June 2012.
- [30] X. Liu, E. K. P. Chong, and N. B. Shroff, "A framework for opportunistic scheduling in wireless networks," *Computer Networks*, vol. 41, pp. 451–474, 2003.
- [31] C. Joo, "On random access scheduling for multimedia traffic in multihop wireless networks with fading channels," *Mobile Computing, IEEE Transactions on*, vol. 12, no. 4, pp. 647–656, April 2013.
- [32] C. Joo, X. Lin, J. Ryu, and N. B. Shroff, "Distributed greedy approximation to maximum weighted independent set for scheduling with fading channels," in *Proceedings of the Fourteenth ACM International Symposium on Mobile Ad Hoc Networking and Computing*, ser. MobiHoc '13. New York, NY, USA: ACM, 2013, pp. 89–98.
- [33] S. Yun, J. Shin, and Y. Yi, "Medium access over time-varying channels with limited sensing cost," *CoRR*, vol. abs/1206.5054, 2012. [Online]. Available: <http://arxiv.org/abs/1206.5054>
- [34] S. Adireddy and L. Tong, "Exploiting decentralized channel state information for random access," *Information Theory, IEEE Transactions on*, vol. 51, no. 2, pp. 537–561, Feb 2005.
- [35] X. Qin and R. Berry, "Exploiting multiuser diversity for medium access control in wireless networks," in *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, vol. 2, March 2003, pp. 1084–1094 vol.2.
- [36] D. Zheng, W. Ge, and J. Zhang, "Distributed opportunistic scheduling for ad hoc networks with random access: An optimal stopping approach," *Information Theory, IEEE Transactions on*, vol. 55, no. 1, pp. 205–222, Jan 2009.
- [37] H. Chen and J. Baras, "Distributed opportunistic scheduling for wireless ad-hoc networks with block-fading model," *Selected Areas in Communications, IEEE Journal on*, vol. 31, no. 11, pp. 2324–2337, November 2013.
- [38] H. Viswanathan and S. Mukherjee, "Performance of cellular networks with relays and centralized scheduling," *IEEE Transactions on Wireless Communications*, vol. 4, no. 5, pp. 2318–2328, Sept 2005.
- [39] D. Giatsios, G. Iosifidis, and L. Tassioulas, "Mobile edge-networking architectures and control policies for 5g communication systems," in *2016 14th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt)*, May 2016, pp. 1–8.

- [40] A. Asadi, Q. Wang, and V. Mancuso, "A survey on device-to-device communication in cellular networks," *IEEE Communications Surveys Tutorials*, vol. 16, no. 4, pp. 1801–1819, Fourthquarter 2014.
- [41] Y.-D. Lin and Y.-C. Hsu, "Multihop cellular: a new architecture for wireless communications," in *Proceedings IEEE INFOCOM 2000. Conference on Computer Communications. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (Cat. No.00CH37064)*, vol. 3, Mar 2000, pp. 1273–1282 vol.3.
- [42] B. Kaufman and B. Aazhang, "Cellular networks with an overlaid device to device network," in *2008 42nd Asilomar Conference on Signals, Systems and Computers*, Oct 2008, pp. 1537–1541.
- [43] K. Doppler, M. Rinne, C. Wijting, C. B. Ribeiro, and K. Hugl, "Device-to-device communication as an underlay to lte-advanced networks," *IEEE Communications Magazine*, vol. 47, no. 12, pp. 42–49, Dec 2009.
- [44] X. Wu, S. Tavildar, S. Shakkottai, T. Richardson, J. Li, R. Laroia, and A. Jovicic, "Flashling: A synchronous distributed scheduler for peer-to-peer ad hoc networks," in *2010 48th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, Sept 2010, pp. 514–521.
- [45] M. H. Hajjesmaili, L. Deng, M. Chen, and Z. Li, "Incentivizing device-to-device load balancing for cellular networks: An online auction design," *IEEE Journal on Selected Areas in Communications*, vol. PP, no. 99, pp. 1–1, 2017.
- [46] S. Swain, N. Mishra, S. Rath, and B. P. S. Sahoo, "Spectrum sharing for d2d communication in 5g cellular networks: An auction-based model," in *2016 IEEE Annual India Conference (INDICON)*, Dec 2016, pp. 1–6.
- [47] N. H. Almofari, S. Kishk, and F. W. Zaki, "Auction based algorithm for distributed resource allocation in multitier-heterogeneous cellular networks," in *2016 11th International Conference on Computer Engineering Systems (ICCES)*, Dec 2016, pp. 426–433.
- [48] R. N. Braithwaite, "Improving data throughput for cell-edge users in a lte network using up-link harq relays," in *Vehicular Technology Conference (VTC Fall), 2011 IEEE*, Sept 2011, pp. 1–5.
- [49] S. Li, E. Ekici, and N. Shroff, "Throughput-optimal queue length based csma/ca algorithm for cognitive radio networks," *IEEE Transactions on Mobile Computing*, vol. 14, no. 5, pp. 1098–1108, May 2015.
- [50] F. P. Kelly, *Reversibility and Stochastic Networks*. New York, NY, USA: Cambridge University Press, 2011.
- [51] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, *Network Flows: Theory, Algorithms, and Applications*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1993.
- [52] M. Salehi Heydar Abad, "Optimal Distributed Scheduling Algorithm for Cooperative Communication Networks," Master's thesis, Sabanci University, 2015.